

REMARKS

Reconsideration of the present application as amended is respectfully requested.

In the Office Action dated November 3, 2003, claims 1-40 were pending. Claims 1-26, 28-30, 32-34, 36-40 were rejected under 35 U.S.C. 103(a) as being unpatentable over Stucka et al. (U.S. Patent No. 5,596,702) in view of Brittain et al. (U.S. 6,195,098). Claims 27, 31, and 35 were rejected under 35 U.S.C. 103(a) as being unpatentable over Stucka in view of Brittan and Kahl et al. (U.S. Patent No. 5,936,625).

A response under 37 C.F.R. 1.116 expedite procedure was filed January 5, 2004 and an advisory action was issued February 4, 2004. In the advisory action, the Examiner stated that the arguments presented did not place the application in condition for allowance.

An RCE is requested herein. In this response, no claim has been cancelled and claims 1, 9, 17, 25, 29, 33, and 37 have been amended. No new matter has been added.

Claims 1-26, 28-30, 32-34, 36-40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stucka in view of Brittan.

In view of the foregoing amendments, Applicant submits that claims 1-40 include limitations not disclosed or suggested by the cited references and they are patentable over the cited references. Specifically, independent claim 1 recites as follows:

1. A method comprising:
extracting a first data from a display buffer, the first data generated by a first application and being associated with a user interface from the first application;
recognizing a layout from the first data; and
using the layout to create an overlay to display a second data generated by a second application, wherein there is no direct link between the first application and the second application, and wherein the first data is extracted

from the display buffer without using an application programming interface (API) of the first application.

(Emphasis added)

Applicant submits that independent claim 1 includes limitations of extracting a first data from a display buffer, where the first data is generated by a first application and being associated with a user interface from the first application and the first data is extracted without using an API of the first application. These limitations are not disclosed or suggested by Stucka and Brittan, individually or in combination.

Rather, Stucka discloses a user interface server (UIS) that provides user interface services to multiple applications in a form of a library via a specific API (application programming interface), similar to a software development kit (SDK), which is used by a developer of the respective application. Specifically, Stucka states:

“A user interface server coupled to applications, a display object store, and a window management system. The user interface server allows an application developer to provide an application with a user interface that is independent of any particular window management system.”

(Abstract of Stucka, emphasis added).

Thus, Stucka discloses how an application communicates with the UIS via a set of APIs, which must be compiled and linked (via a compiler and linker of a software development tool, such as a C/C++ compiler and linker) with the application when the application is developed by the developer, to access one or more functions provided by the UIS. See, for example, col. 20, line 33 to col. 21, line 40, and col. 22, line 58 to col. 23, line 55. However, it is respectfully submitted that Stucka does not disclose the limitation of extracting data of an application from a display buffer, particularly, without using an API of

the application. In fact, there is no mention of extracting data from a display buffer in Stucka or Brittain.

Even if, for the sake of argument, the access from an application to a UIS server may be considered as “extracting data” from a first application (e.g., UIS server is considered as a first application), the access to the UIS cannot be performed via a display buffer. As shown in Figure 3 of Stucka, the UIS is a server that is a dedicated server providing specific services over a network connection (e.g., connection 56 of Figure 3). One skilled in the art would not attempt to store, even if it were possible, the UIS (e.g., a server), specifically the library, such as display object store 46, in a display buffer accessed by another application, such as application 50. Applicant submits that one with ordinary skill in the art would not believe a server or a library, such as display object store 46 of Stucka, would reasonably be stored in a display buffer.

UIS 48 of Stucka is not stored in a display buffer 72. Rather, UIS 48, as well as other components, such as working memory area 78, display object store 46, and operating system 74, etc., are stored in a general purpose RAM 38. If Stucka intended to have UIS 48 operating within a display buffer, UIS 48 would have been shown within display buffer 72. Even if the communications between applications (50, 53, and 54) and UIS 48 can be considered as extracting data from another application. Such an extraction operation is not performed from a display buffer, particularly display buffer 72. At most, it can only be considered an extraction from RAM 38. See, for example, Fig. 2, col. 7, line 47 to col. 8, line 25 of Stucka. Given the fact that Stucka fails to disclose or suggest that such operations be performed within display buffer 72, it is respectfully submitted that those with ordinary skill in the art would not consider, based on the teachings of Stucka, that the above operations can be performed within a display buffer.

Further, even if the access from an application to a UIS server may be considered as “extracting data” from a first application (e.g., UIS server is considered as a first application), the access to the UIS cannot be performed without using an API of the UIS. As described above, in order to access UIS, an application has to be compiled and linked with the APIs of the UIS. Without using the APIs of the UIS, an application cannot access the UIS.

The Examiner stated that Brittain discloses a display buffer and it would be obvious to combine Stucka and Brittain. Applicant respectfully disagrees and for the reasons discussed above, Applicant respectfully submits that, in contrast, a person of ordinary skill in the art would not combine Stucka and Brittain, because such combination lacks reasonable successes, as discussed above.

In addition, independent claim 1 includes a limitation of recognizing a layout from the first data extracted. Applicant submits that this limitation is also absent from the cited references, individually or in combination. Nowhere in Stucka or Brittain is there disclosure or a suggestion recognizing a layout from the data extracted from a display buffer, which is generated by another application. In fact, there is no mention of recognizing a layout in a display buffer in the cited references, individually or in combination. As discussed above, Stucka relies on a set of APIs used by an application to access a server to retrieve user interface data. There is no need to recognize a layout from the data provided by the UIS, particularly using a pattern recognition operation as recited in claim 2. The respective application has to be compiled and linked with the corresponding API, such as the header files and libraries of the UIS, during the development of the application. At run time, when the application calls a specific function (e.g., an API function) to access the server, the server provides services to the application. It is irrelevant whether the application would recognize a layout of the data returned from the server because the application has to call the

corresponding function exported and published by the server in a format required by the API, regardless of whether the application recognizes the layout.

In contrast, the present invention as claimed recognizes a layout of a user interface generated by another application in a display buffer (e.g., raster data that is about to be displayed or currently displayed) and overlays its data with the recognized interface, without using an API of the application generating the user interface. Therefore, one would not be motivated, based on the teachings of Stucka, to arrive at the present invention as claimed.

Furthermore, independent claim 1 further includes a limitation of using the recognized layout to create an overlay to display a second data generated by a second application, wherein there is no direct link between the first and second applications (e.g., without requiring communications between the first and second applications via APIs). Applicant submits that this limitation is also absent from the cited references, individually or in combination. Applicant submits that Stucka fails to disclose or suggest using the recognized layout from a first application to create an overlay to display data from a second application. As discussed above, an application of Stucka has to rely on a set of APIs to access the UIS in order to create a user interface. Nowhere in the cited references disclose or suggest an operation of creating an overlay of a recognized layout, when an application accesses the UIS via a set of APIs. There is no need or motive to create an overlay to display the data from another application, as discussed above.

Further, contrary to the present invention as claimed, Applicant submits that there is a direct link between the application and the UIS in Stucka. The source code of an application in Stucka must be compiled and linked with the exported or published APIs from the UIS. Any changes in the APIs would cause the communications between the application and the UIS to fail, which requires either the application or the UIS, or the both to be recompiled and

linked. The applications and the UIS of Stucka depend on a set of APIs mutually agreed upon, which teaches away from the present invention as claimed. See, for example, col. 23, lines 38 to 54. Therefore, there is a direct link between an application and the UIS in Stucka. In contrast, there is no direct link between the first and second applications in the present invention as claimed, where the data is extracted without using an API of the first application.

Some of the above remarks have been provided in Applicant's previous response filed September 2, 2003. It appears that the Examiner fails to consider all of the remarks previously filed. In response to Applicant's previous response filed September 2, 2003, the Examiner stated:

"Stucka extracts a first data from a display buffer (Fig 2, col 8, lines 8-26). Furthermore, Stucka recognizes a lay out from the first data (col 23, lines 62-67, col 24, lines 37-60). The examiner infers to the fact that command parameter data required to initialize user interface components such as background and foreground color are part of the layout pattern from the first data. Although Stucka doesn't teach extracting the data from a video card, by combining Stucka with Brittain would allow Stucka to meet this limitation:

Stucka teaches extracts a first data from a display buffer (Fig 2, col 8, lines 8-26). At the time when Stucka's reference was filed it was common for the video card to have its display buffer on the RAM of the computer. However, now it is more common for the video card to have its own memory to store the video buffer. Therefore, it would be inherent for the system to extract the data from the display buffer of the video card."

(11/3/2003 Office Action, page 13, emphasis added).

Applicant respectfully disagrees. As discussed above, the system RAM of Stucka is not and cannot be considered as a display buffer. Specifically, the section of Stucka relied upon by the Examiner states:

"A Working memory area 78 is also shown in RAM 38. The Working Memory Area 78 can be utilized by any of the elements shown in RAM 38. The working memory area can be utilized by the applications (50, 53, 54), window management system 58, user interface server 48, OS 74 and other functions. The working memory area 78 may be partitioned amongst the elements and within an element. The working memory area 78 may be utilized for communication, buffering, temporary storage, or storage of data while a program is running. For instance, the user interface server 48

utilizes the working memory area 78 for loading and operating on user interfaces retrieved from the display object store 46. The working memory area 78 may also be utilized by the operating system for multi-tasking purposes. The working memory area 78 can be included as part of the application or it may be located in a common area. In either case there is no requirement that the working memory be contiguous in RAM 38."

(Stucka, Fig. 2, col. 8, lines 8-26, emphasis added).

Applicant respectfully submits that the above section does not use a display buffer, such as display buffer 72 shown in Fig. 2 of Stucka. Clearly, Stucka's RAM 38 and working memory area 78 are not a display buffer. Otherwise, the above operations, which interpreted by the Examiner as extracting data from a display buffer, would be performed within display buffer 72 of Stucka (see, Fig. 2). Applicant respectfully submits that a display buffer is a dedicated memory only used by a display adapter and a display driver. Whether the display buffer is on a RAM is irrelevant. Given the fact that Stucka fails to disclose or suggest such operations be performed within display buffer 72, it is respectfully submitted that Stucka teaches away from the present invention as claimed and those with ordinary skill in the art would not consider, based on the teachings of Stucka, that the above operations can be performed within a display buffer.

It appears that the Examiner considers RAM 38 or working memory area is a display buffer. Applicant respectfully disagrees. Applicant respectfully submits that those with ordinary skill in the art would not consider RAM 38 is a display buffer for storing user interface server 48, window system I/F 70, window management system 58, display object store 46, operating system 74, etc. Applicant respectfully challenges the Examiner to provide references to support the Examiner's interpretation of a display buffer for storing user interface server 48, window system I/F 70, window management system 58, display object store 46, operating system 74, etc.

Therefore, for the reasons discussed above, independent claim 1 is patentable over the cited references. Similarly, independent claims 9, 17, 25, 29, 33, and 37 include limitations similar to those discussed above. Therefore, for reasons similar to those discussed above, independent claims 9, 17, 25, 29, 33, and 37 are patentable over the cited references.

The rest of the claims depend on one of the above independent claims, thus include all of the distinct features of the respective independent claim, and therefore, for the reasons similar to those discussed above, are patentable over the cited references. Withdrawal of the rejections is respectfully submitted.


In view of the foregoing, Applicant respectfully submits the present application is now in condition for allowance. If the Examiner believes a telephone conference would expedite or assist in the allowance of the present application, the Examiner is invited to call the undersigned attorney at (408) 720-8300.

Please charge Deposit Account No. 02-2666 for any shortage of fees in connection with this response.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Date: 3/2, 2004


Kevin G. Shao
Reg. No. 45,095

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025-1026
(408) 720-8300